Name: Muhammad Naufal Faza

ID Number: 2006577252

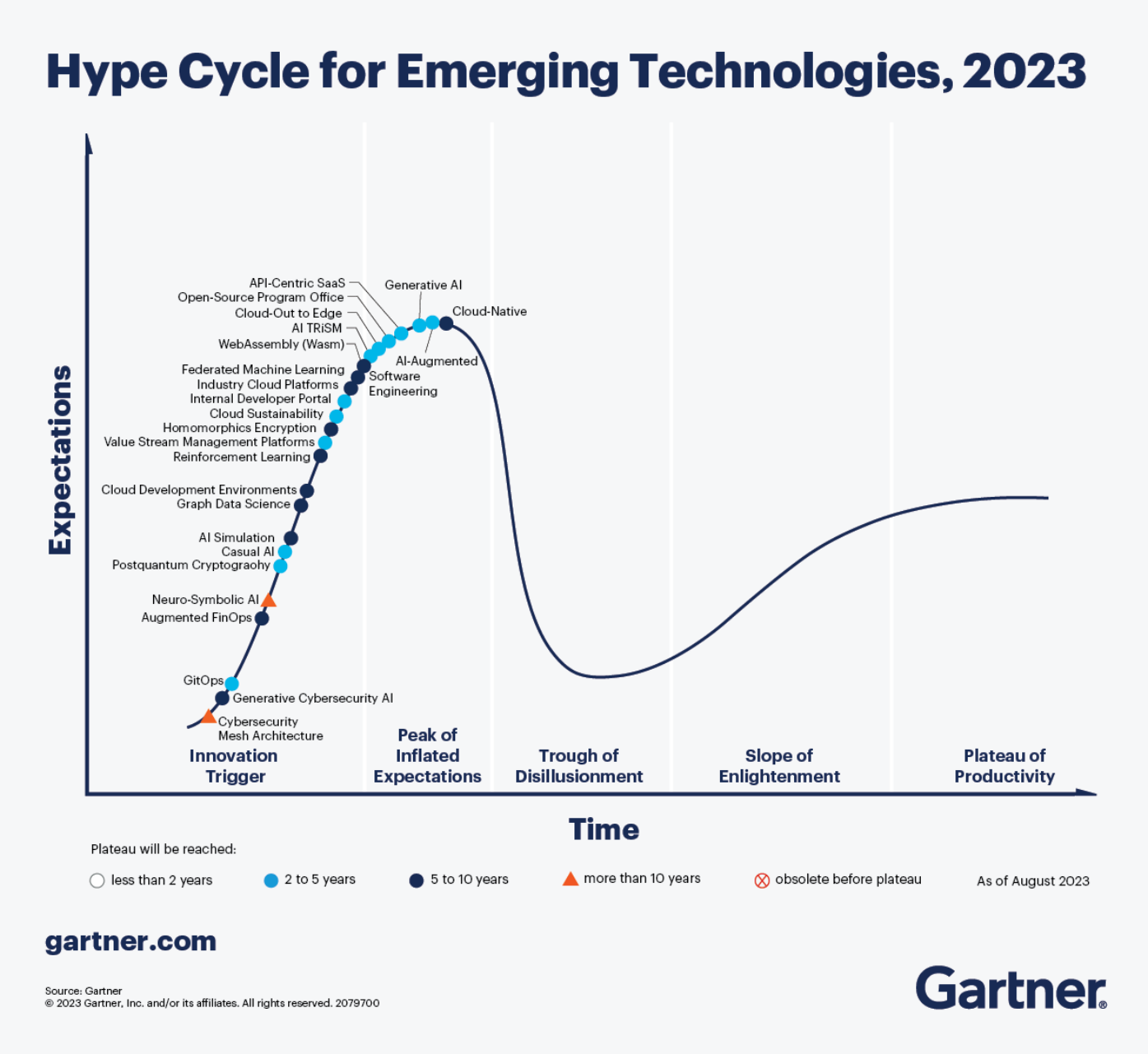
Class: Software Architecture for Digital Enterprise

GARTNER HYPE CYCLE FOR EMERGING TECHNOLOGIES 2023, SOFTWARE ENGINEERING, & ENTERPRISE SOFTWARE ARCHITECTURE

Introduction

Overview of Gartner Hype Cycle

The Gartner Hype Cycle is a graphical tool that illustrates the maturity and adoption of emerging technologies over time. It helps organizations understand the risks and opportunities of investing in new innovations, and guides them to make informed decisions about when and how to adopt them. The Hype Cycle consists of five phases: innovation trigger, peak of inflated expectations, trough of disillusionment, slope of enlightenment, and plateau of productivity.



Importance of Tracking Emerging Technologies

Emerging technologies are disruptive by nature and can have profound impacts on business and society. They can create new markets, transform existing ones, or render them obsolete. They can also enable new capabilities, enhance customer experiences, improve efficiency, and reduce costs. However, emerging technologies also come with challenges and uncertainties, such as technical feasibility, ethical implications, regulatory compliance, and social acceptance. Therefore, it is essential for organizations to track the trends and trajectories of emerging technologies and evaluate their potential benefits and risks.

Four Key Themes in the 2023

The 2023 Gartner Hype Cycle for Emerging Technologies identifies 25 must-know technologies that are expected to have significant influence on business and society in the next two to 10 years. These technologies are grouped into four key themes: emergent AI, developer experience, pervasive cloud, and human-centric security and privacy. Each theme represents a major shift in how technology is used and delivered, and reflects the changing needs and expectations of customers, employees, and stakeholders. Writer will explore each theme in detail and discuss how they can help organizations achieve digital transformation.

Theme 1: Emergent AI

AI Simulation

AI simulation is the process of using artificial intelligence (AI) techniques to create, run, and analyze models of complex systems or phenomena. AI simulation can help researchers and practitioners understand the behavior, dynamics, and outcomes of real-world situations that are difficult or impossible to observe or experiment with directly. AI simulation can also enable the exploration of alternative scenarios, the optimization of parameters, the generation of novel solutions, and the discovery of new insights. AI simulation can be applied to various domains, such as physics, biology, engineering, economics, social sciences, and more.

In recent years, AI simulation has witnessed several breakthroughs that have pushed the boundaries of what is possible with this technology. Some examples are:

- DALL-E, an AI system that can create realistic images from text descriptions. For instance, given the prompt “a dragon made of flowers”, DALL-E can produce an image that matches the description.
- ChatGPT, an AI system that can generate coherent and fluent text and code from text prompts. For example, given the prompt “write a poem about love”, ChatGPT can produce a poem that follows the rhyme and meter rules.
- AlphaCode, an AI system that can solve complex coding problems better than most human coders. For example, AlphaCode can write a program that sorts a list of numbers in ascending order using any programming language.
- AI-enriched Simulation, a technology that uses AI to identify the most promising simulations to run on a massive data set and to determine the best computing infrastructure for the task. For example, AI-enriched Simulation can help researchers discover new molecules or design new chips faster and more efficiently.

AI simulation has a wide range of potential applications in various fields and industries. Some examples are:

- Business and Finance: AI simulation can help businesses and financial institutions improve their decision-making, forecasting, risk management, and customer service by modeling market trends, customer behavior, economic scenarios, and more.
- Healthcare: AI simulation can help healthcare providers and researchers improve diagnosis, treatment, prevention, and drug discovery by modeling diseases, symptoms, drugs, vaccines, and more.
- Agriculture: AI simulation can help farmers and agronomists improve crop yield, quality, and resilience by modeling soil conditions, weather patterns, pests, diseases, and more.
- Smart Cities: AI simulation can help urban planners and managers improve traffic flow, energy efficiency, public safety, and environmental quality by modeling transportation networks, power grids, buildings, pollution sources, and more.
- Cybersecurity: AI simulation can help cybersecurity experts and analysts improve their defense and attack strategies by modeling cyber threats, vulnerabilities, attacks, and countermeasures.

Causal AI

Causal AI is a branch of artificial intelligence that aims to discover and exploit the causal relationships between variables, rather than just the statistical correlations. Causal AI can answer questions such as "What is the effect of X on Y?" or "How can we change Y by intervening on X?" that are beyond the reach of traditional machine learning methods. Causal AI is based on the theory of causality developed by Judea Pearl and others, which uses graphical models called causal diagrams to represent and reason about the causal structure of a system. Causal AI can also incorporate domain knowledge, counterfactual reasoning, and causal inference techniques to learn from both observational and experimental data.

Causal AI has many potential applications in various fields and industries, where understanding the cause-and-effect relationships is crucial for decision making, policy making, and innovation. Some examples are:

- Business and Finance: Causal AI can help businesses and financial institutions improve their decision making, forecasting, risk management, and customer service by modeling market trends, customer behavior, economic scenarios, and more For example, causaLens, a company that provides causal AI solutions, helps hedge funds and asset managers to discover the true drivers of market movements and optimize their trading strategies.
- Healthcare: Causal AI can help healthcare providers and researchers improve diagnosis, treatment, prevention, and drug discovery by modeling diseases, symptoms, drugs, vaccines, and more For example, ZIF, a company that provides AI-powered healthcare solutions, uses causal AI to identify the root causes of patient deterioration and provide personalized care recommendations.
- Agriculture: Causal AI can help farmers and agronomists improve crop yield, quality, and resilience by modeling soil conditions, weather patterns, pests, diseases, and more For example, Senslytics, a company that provides IoT and AI solutions for industrial applications, uses causal AI to interpret rare events in oil and gas upstream operation.
- Smart Cities: Causal AI can help urban planners and managers improve traffic flow, energy efficiency, public safety, and environmental quality by modeling transportation networks, power grids, buildings, pollution sources, and more For example, Xplain Data, a company that provides explainable AI solutions for smart cities, uses causal AI to optimize traffic signal timing and reduce congestion.

Causal AI is a rapidly evolving field that promises to revolutionize the way we use data and technology to understand and shape the world. Causal AI has the potential to overcome some of the limitations and challenges of current AI approaches, such as data scarcity, data quality, data bias, model interpretability, model robustness, model generalization, model ethics, and model accountability. Causal AI can also enable new capabilities and opportunities for innovation, such as scenario planning, what-if analysis, hypothesis testing, intervention design, policy evaluation, and causal discovery. Gartner, a leading research and advisory firm, has listed causal AI as one of the five critical technologies in accelerated AI automation in its 2023 Hype Cycle for Emerging Technologies report. According to Gartner, causal AI will reach mainstream adoption in the next two to five years.

Federated Learning

Federated learning is a machine learning technique that enables organizations to train AI models on decentralized data, without the need to centralize or share that data. This means businesses can use AI to make better decisions without sacrificing data privacy and risking breaching personal information. Federated learning works by distributing the model training process across multiple devices or nodes, such as mobile phones, laptops, or servers, that hold their own local data. Each node trains a local model on its own data and periodically communicates with a central server to update a global model. The communication only involves the model parameters, such as weights and biases, not the raw data. Therefore, the data remains local and private to each node.

Federated learning has several benefits over traditional machine learning methods that require centralized data. Some of these benefits are:

- Data privacy: Federated learning preserves the privacy of the data owners by keeping the data local and only exchanging model parameters. This reduces the risk of data leakage, theft, or misuse by unauthorized parties
- Data security: Federated learning enhances the security of the data by avoiding data transfer over networks that may be vulnerable to attacks or breaches. This also reduces the bandwidth and storage costs associated with data transmission and aggregation
- Data quality: Federated learning leverages the diversity and richness of the data distributed across multiple nodes, which may capture different aspects of the real-world phenomena. This can improve the accuracy and generalization of the global model

However, federated learning also faces several challenges that need to be addressed. Some of these challenges are:

- Data heterogeneity: Federated learning deals with data that may be heterogeneous in terms of distribution, size, quality, and availability across different nodes. This can lead to uneven or biased learning outcomes and affect the convergence and stability of the global model
- Communication efficiency: Federated learning requires frequent communication between the nodes and the central server to synchronize the model updates. This can incur high latency and overhead, especially when the number of nodes is large or the network is unreliable
- Privacy and security threats: Federated learning still exposes some privacy and security risks, such as inference attacks that can reveal sensitive information from the model parameters, or poisoning attacks that can compromise the integrity of the global model by injecting malicious updates from compromised nodes.

Federated learning has been adopted by various industries and domains that deal with sensitive or large-scale data. Some examples are:

- Healthcare: Federated learning can help healthcare providers and researchers improve diagnosis, treatment, prevention, and drug discovery by modeling diseases, symptoms, drugs, vaccines, and more, while respecting the privacy and confidentiality of patient data For example, Owkin, a company that provides AI solutions for medical research, uses federated learning to enable collaboration among hospitals and researchers without sharing patient data.
- Manufacturing: Federated learning can help manufacturers improve product quality, efficiency, and safety by modeling machine conditions, performance, faults, and maintenance, while protecting their proprietary data and intellectual property For example, Senslytics, a company that provides IoT and AI solutions for industrial applications, uses federated learning to interpret rare events in oil and gas upstream operation.
- Telecommunications: Federated learning can help telecommunication companies improve network optimization, customer experience, and service innovation by modeling user behavior, preferences, feedback, and usage patterns, while preserving user privacy and reducing data transmission costs For example, Google, a leading technology company, uses federated learning to personalize keyboard suggestions on Android devices without uploading user data to the cloud.

Graph Data Science

Graph data science is a field that combines graph theory, graph algorithms, and graph analytics to extract insights and value from complex data that can be represented as graphs. Graphs are mathematical structures that consist of nodes (or vertices) and edges (or links) that connect them. Graphs can capture the relationships, interactions, and dependencies among entities in various domains, such as social networks, biological networks, transportation networks, knowledge graphs, and more. Graph data science can help discover patterns, identify anomalies, measure similarities, infer causality, optimize outcomes, and generate recommendations from graph data.

Graph data science has many practical applications in various fields and industries, where understanding the structure and dynamics of complex systems is crucial for decision making, problem solving, and innovation. Some examples are:

- Social Media: Graph data science can help social media platforms improve user engagement, content moderation, community detection, and influencer analysis by modeling user behavior, preferences, feedback, and social connections. For example, Facebook, a leading social media company, uses graph data science to power its News Feed ranking algorithm, which determines what content to show to each user based on their interests and interactions.
- Healthcare: Graph data science can help healthcare providers and researchers improve diagnosis, treatment, prevention, and drug discovery by modeling diseases, symptoms, drugs, vaccines, and more. For example, Neo4j, a company that provides graph database solutions, uses graph data science to enable collaboration among hospitals and researchers without sharing patient data.
- E-commerce: Graph data science can help e-commerce platforms improve product discovery, recommendation, personalization, and fraud detection by modeling product features, customer behavior, purchase history, and

reviews. For example, Amazon, a leading e-commerce company, uses graph data science to power its product recommendation engine, which suggests relevant items to each customer based on their browsing and buying patterns.

- Cybersecurity: Graph data science can help cybersecurity experts and analysts improve their defense and attack strategies by modeling cyber threats, vulnerabilities, attacks, and countermeasures. For example, CyberCube, a company that provides cyber risk analytics solutions for the insurance industry, uses graph data science to assess the cyber risk exposure of organizations based on their network topology and security posture.

Graph data science is a rapidly evolving field that promises to unlock new possibilities and opportunities for data-driven innovation. Some of the emerging trends in graph data science are:

- Graph Neural Networks: Graph neural networks are deep learning models that can learn from graph data by encoding the node features and edge information into latent representations that can be used for various tasks such as node classification, link prediction, graph generation, and graph embedding. Graph neural networks can leverage the power of deep learning to handle large-scale and dynamic graph data with complex attributes and relations.
- Knowledge Graphs: Knowledge graphs are large-scale graphs that represent facts and concepts from various domains in a structured and semantic way. Knowledge graphs can integrate heterogeneous and multi-modal data sources into a unified framework that can support natural language processing (NLP), computer vision (CV), machine learning (ML), and artificial intelligence (AI) applications. Knowledge graphs can enable intelligent reasoning, question answering (QA), information extraction (IE), entity linking (EL), and natural language generation (NLG) from graph data.
- Graph Databases: Graph databases are specialized databases that store and query graph data in a native way. Graph databases can offer high performance, scalability, flexibility, and expressiveness for graph data management and analysis. Graph databases can support various graph operations such as traversal (finding paths or neighbors), pattern matching (finding subgraphs or motifs), aggregation (finding statistics or metrics), transformation (modifying or updating graphs), and visualization (displaying graphs). Graph databases can enable real-time and interactive graph data science applications.

Neuro-symbolic AI

Neuro-symbolic AI is a field that aims to combine the strengths of neural networks and symbolic systems to create more powerful and intelligent artificial agents. Neural networks are machine learning models that can learn from data and perform tasks such as perception, recognition, prediction, and generation. Symbolic systems are logic-based models that can represent and manipulate abstract concepts, rules, and reasoning. Neuro-symbolic AI can leverage the advantages of both approaches, such as the ability to handle uncertainty, ambiguity, and complexity from neural networks, and the ability to explain, generalize, and transfer knowledge from symbolic systems.

Hybrid AI systems are systems that integrate neural networks and symbolic systems in various ways to achieve neuro-symbolic AI. There are different types of hybrid AI systems, depending on how they combine the two paradigms. Some examples are:

- Neural-Symbolic Integration: These systems use neural networks to learn symbolic representations or rules from data, or use symbolic systems to guide or constrain neural network learning. For example, NeuroLogic is a system that uses neural networks to induce logic programs from examples.
- Neural-Symbolic Communication: These systems use neural networks to encode or decode symbolic information for communication or interaction with other agents or humans. For example, Neural State Machines is a system that uses neural networks to parse natural language queries into symbolic expressions and execute them on a knowledge graph.
- Neural-Symbolic Computation: These systems use neural networks to perform symbolic computations or operations on data or knowledge. For example, Neural Theorem Provers is a system that uses neural networks to prove mathematical theorems by applying logical rules.

Neuro-symbolic AI is a promising field that has the potential to advance the state-of-the-art in artificial intelligence and enable new applications and capabilities. Some of the potential benefits of neuro-symbolic AI are:

- Explainability: Neuro-symbolic AI can provide explanations for the decisions or actions of artificial agents by using symbolic representations or rules that are understandable by humans. This can enhance the trust, transparency, and accountability of AI systems.
- Generalization: Neuro-symbolic AI can improve the generalization and robustness of artificial agents by using symbolic systems to transfer knowledge across domains or tasks. This can reduce the data and computational requirements of AI systems.
- Creativity: Neuro-symbolic AI can enhance the creativity and innovation of artificial agents by using symbolic systems to generate novel solutions or concepts. This can enable AI systems to solve complex problems or produce original works.

Reinforcement Learning

Reinforcement learning is a machine learning technique that enables an agent to learn from its own actions and experiences in an environment. Reinforcement learning is based on the idea of trial-and-error learning, where the agent tries different actions and receives feedback in the form of rewards or penalties. The goal of reinforcement learning is to find the optimal policy that maximizes the expected cumulative reward over time. Reinforcement learning consists of four main components: an agent, an environment, a set of actions, and a reward function.

Reinforcement learning has many notable use cases in various fields and industries, where the agent can learn to perform complex tasks or optimize outcomes without explicit supervision or guidance. Some examples are:

- Games: Reinforcement learning can help agents learn to play games at superhuman levels by exploring and exploiting different strategies and tactics. For example, AlphaGo is an agent that uses reinforcement learning to master the game of Go, one of the most challenging board games in the world.
- Robotics: Reinforcement learning can help robots learn to perform various skills and behaviors by interacting with their surroundings and adapting to changing conditions. For example, OpenAI is a research organization that uses reinforcement learning to teach robots how to manipulate objects, such as solving a Rubik’s cube.
- Healthcare: Reinforcement learning can help healthcare providers and researchers improve diagnosis, treatment, prevention, and drug discovery by modeling diseases, symptoms, drugs, vaccines, and more. For example,

DeepMind is a company that uses reinforcement learning to design new proteins that can enhance the efficacy of vaccines.

Reinforcement learning is a promising field that has the potential to advance the state-of-the-art in artificial intelligence and enable new applications and capabilities. Some of the future possibilities of reinforcement learning are:

- Multi-agent Systems: Multi-agent systems are systems that consist of multiple agents that interact with each other and their environment. Multi-agent systems can enable complex and emergent phenomena, such as cooperation, competition, coordination, communication, and negotiation. Reinforcement learning can help agents learn to cooperate or compete with other agents in various scenarios, such as social dilemmas, market mechanisms, or collective decision making.
- Lifelong Learning: Lifelong learning is the ability of an agent to continuously learn from new data and experiences throughout its lifetime. Lifelong learning can enable agents to adapt to changing environments, transfer knowledge across domains or tasks, and overcome catastrophic forgetting. Reinforcement learning can help agents learn to balance exploration and exploitation, update their beliefs and policies, and retain relevant information over time.
- General Intelligence: General intelligence is the ability of an agent to perform any intellectual task that a human can do. General intelligence can enable agents to reason, plan, learn, communicate, and create across various domains and contexts. Reinforcement learning can help agents learn to achieve general intelligence by integrating different cognitive abilities, such as perception, memory, attention, language, logic, and creativity.

Theme 2: Developer Experience

Value Stream Management Platform

A Value Stream Management Platform (VSMP) is a software tool that helps organizations manage and optimize their value streams, which are the series of activities and processes that deliver value to customers. A VSMP enables organizations to visualize, measure, and improve the flow of value from ideation to delivery, across different stages, teams, and tools. A VSMP also helps organizations align their value streams with their strategic goals, customer needs, and market demands. A VSMP plays a vital role in software development, as it helps developers and stakeholders collaborate, communicate, and coordinate more effectively and efficiently.

A VSMP provides several benefits for software development teams, such as:

- Increased visibility: A VSMP provides a comprehensive and real-time view of the status, progress, and performance of the value streams, as well as the dependencies, bottlenecks, and risks that affect them. This helps teams identify and resolve issues, prioritize tasks, and make informed decisions.
- Enhanced collaboration: A VSMP facilitates cross-functional and cross-organizational collaboration by integrating different tools, platforms, and data sources that are used by the teams. This helps teams share information, feedback, and best practices, and coordinate their actions.
- Improved quality: A VSMP supports continuous improvement and innovation by enabling teams to monitor and analyze the quality and value of their products and services. This helps teams detect and prevent defects, ensure compliance, and deliver customer satisfaction.

- Reduced waste: A VSMP eliminates waste and inefficiency by streamlining and automating the value streams, as well as applying lean and agile principles. This helps teams reduce costs, save time, and increase productivity.

VSMP adoption is a growing trend in the software industry, as more organizations recognize the need and value of value stream management. According to a report by Forrester Research, the global market for VSMP is expected to reach $2.3 billion by 2023, with a compound annual growth rate of 40%. The report also identifies some of the key drivers and challenges for VSMP adoption. Some of the drivers are:

- The increasing complexity and diversity of software development environments, which require more integration, coordination, and governance.
- The rising demand for faster delivery and higher quality of software products and services, which require more visibility, measurement, and optimization.
- The growing awareness and maturity of value stream management practices and methodologies, which provide more guidance, support, and benefits.

Some of the challenges are:

- The lack of standardization and interoperability among different tools, platforms, and data sources that are used by the teams.
- The difficulty of changing the culture and mindset of the organization to embrace value stream management principles and values.
- The scarcity of skills and expertise in value stream management among the teams.

AI-augmented Software Engineering

AI-augmented software engineering is an emerging field that applies artificial intelligence (AI) techniques and tools to assist and enhance software development processes and outcomes. AI-augmented software engineering can leverage the capabilities of AI, such as learning, reasoning, planning, and creativity, to automate, optimize, or improve various aspects of software engineering, such as requirements analysis, design, coding, testing, debugging, deployment, maintenance, and evolution. AI-augmented software engineering can also enable new forms of human-AI collaboration and co-creation, where software developers and AI agents work together to produce high-quality software products and services.

AI-augmented software engineering has many case studies in various domains and industries, where AI techniques and tools can help software developers solve complex problems or achieve better results. Some examples are:

- Natural Language Processing (NLP): NLP is a branch of AI that deals with the analysis and generation of natural language texts or speech. NLP can help software developers create or improve software applications that involve natural language interaction, such as chatbots, voice assistants, text summarizers, sentiment analyzers, and more. For example, ChatGPT is an AI system that can generate coherent and fluent text and code from text prompts. For example, given the prompt “write a poem about love”, ChatGPT can produce a poem that follows the rhyme and meter rules.
- Computer Vision (CV): CV is a branch of AI that deals with the analysis and generation of visual information, such as images or videos. CV can help software developers create or improve software applications that involve

visual processing, such as face recognition, object detection, image editing, video synthesis, and more. For example, DALL-E is an AI system that can create realistic images from text descriptions. For example, given the prompt “a dragon made of flowers”, DALL-E can produce an image that matches the description.

- Automated Programming (AP): AP is a branch of AI that deals with the synthesis or optimization of computer programs or code. AP can help software developers create or improve software applications that involve programming tasks, such as code generation, code completion, code refactoring, code debugging, code testing, and more. For example, AlphaCode is an AI system that can solve complex coding problems better than most human coders. For example, AlphaCode can write a program that sorts a list of numbers in ascending order using any programming language.

AI-augmented software engineering is a promising field that has the potential to transform the way software is developed and delivered. AI-augmented software engineering can provide several benefits for software developers and stakeholders, such as:

- Increased productivity: AI-augmented software engineering can increase the productivity and efficiency of software developers by automating or simplifying tedious or repetitive tasks, such as coding, testing, debugging, or documentation. This can reduce the time and cost of software development and delivery.
- Enhanced quality: AI-augmented software engineering can enhance the quality and reliability of software products and services by detecting or preventing errors, bugs, or defects, as well as ensuring compliance with standards or specifications. This can improve the performance and satisfaction of software users and customers.
- Improved creativity: AI-augmented software engineering can improve the creativity and innovation of software developers by generating or suggesting novel solutions or concepts, as well as enabling new forms of expression or interaction. This can enable software developers to solve complex problems or produce original works.

However, AI-augmented software engineering also poses some challenges and risks that need to be addressed. Some of these challenges are:

- Ethical issues: AI-augmented software engineering raises ethical issues such as accountability, responsibility, transparency, fairness, privacy, and security. Who is responsible for the actions or outcomes of AI agents? How can we ensure that AI agents follow ethical principles or values? How can we protect the data or information used by AI agents? How can we prevent or mitigate the misuse or abuse of AI agents?
- Social issues: AI-augmented software engineering raises social issues such as trust, acceptance, collaboration, and co-creation. How can we trust the decisions or actions of AI agents? How can we accept the role or influence of AI agents? How can we collaborate or co-create with AI agents? How can we balance human-AI interaction?
- Technical issues: AI-augmented software engineering raises technical issues such as complexity, diversity, scalability, and adaptability. How can we handle the complexity and diversity of software development environments? How can we scale up or scale out AI techniques and tools? How can we adapt to changing requirements or conditions?

Understanding API-centric SaaS

API-centric SaaS (Software as a Service) is a software delivery model that provides software applications as a service through application programming interfaces (APIs). APIs are standardized and documented interfaces that enable

communication and integration between different software components or systems. API-centric SaaS allows users to access, consume, and customize software applications through APIs, without the need to install, maintain, or update them on their own devices or servers. API-centric SaaS also enables developers to build, deploy, and scale software applications faster and easier by using APIs as building blocks or Lego pieces.

API-centric SaaS provides several advantages for businesses, such as:

- Cost-effectiveness: API-centric SaaS reduces the upfront and operational costs of software development and maintenance by eliminating the need to purchase, install, or manage hardware or software infrastructure. API-centric SaaS also allows users to pay only for what they use, based on the number of API calls or transactions.
- Flexibility: API-centric SaaS offers more flexibility and customization options for users by allowing them to choose, combine, and modify different software features or functionalities through APIs. API-centric SaaS also allows users to integrate different software applications or systems with each other through APIs, creating a seamless and unified user experience.
- Scalability: API-centric SaaS enables more scalability and reliability for software applications by allowing them to handle variable and unpredictable workloads or demands through APIs. API-centric SaaS also allows software applications to leverage the cloud computing resources and capabilities of the service providers, such as storage, processing, security, or backup.

API-centric SaaS is a growing trend in the software industry, as more businesses and consumers recognize the value and potential of APIs. According to a report by Gartner, the global market for API management solutions is expected to reach $6.2 billion by 2023, with a compound annual growth rate of 28.1%. The report also identifies some of the key drivers and challenges for API-centric SaaS adoption. Some of the drivers are:

- The increasing demand for digital transformation and innovation, which require more integration, automation, and collaboration among different software applications or systems.
- The rising popularity of microservices and serverless architectures, which rely on APIs to decompose and distribute software applications into smaller and independent units.
- The growing awareness and maturity of API design and development practices and standards, which provide more guidance, support, and benefits for API-centric SaaS.

Some of the challenges are:

- The complexity and diversity of API management tasks and processes, such as design, documentation, testing, security, monitoring, analytics, or governance.
- The difficulty of changing the culture and mindset of the organization to embrace API-centric SaaS principles and values.
- The scarcity of skills and expertise in API-centric SaaS among the developers and users.

Gitops

GitOps is a software development and delivery paradigm that uses Git as the single source of truth and the central platform for managing, deploying, and operating software applications. GitOps leverages the features and benefits of Git, such as

version control, collaboration, code review, and automation, to enable faster, safer, and more reliable software delivery. GitOps follows four main principles:

1. Declarative configuration: The desired state of the software application and its environment is defined and stored in Git as code, such as YAML or JSON files. This allows for easy tracking, auditing, and replication of the configuration.
2. Immutable infrastructure: The software application and its environment are treated as immutable artifacts that are built once and deployed many times, without any changes or modifications. This ensures consistency and reliability across different stages and environments.
3. Continuous delivery: The software application and its environment are continuously delivered to the target environment using automated pipelines that are triggered by Git events, such as commits or merges. This enables faster feedback loops and shorter time to market.
4. Observability: The actual state of the software application and its environment is continuously monitored and compared with the desired state stored in Git. Any discrepancies or deviations are detected and reported, and can be automatically or manually corrected.

GitOps can be implemented in various ways using different tools and platforms, depending on the specific needs and preferences of the software development teams. However, a common workflow for GitOps is as follows:

- The developers write code for the software application and push it to a Git repository. The code is reviewed and tested by other developers or tools before being merged to the main branch.
- The configuration files for the software application and its environment are also stored in a separate Git repository. The configuration files specify the parameters, dependencies, resources, and policies for the deployment and operation of the software application.
- A continuous integration (CI) tool builds the software application from the code repository and creates a container image that is pushed to a container registry. The CI tool also updates the configuration repository with the latest image tag or version number.
- A continuous delivery (CD) tool monitors the configuration repository for any changes and deploys the software application to the target environment using the configuration files. The CD tool also verifies that the deployment is successful and meets the desired state.
- A monitoring tool collects metrics and logs from the software application and its environment and compares them with the desired state stored in the configuration repository. The monitoring tool alerts the developers or operators if there are any issues or anomalies that need to be addressed.

GitOps is a promising trend in the DevOps movement, as it offers several advantages for software development teams, such as:

- Improved collaboration: GitOps fosters collaboration among developers and operators by using Git as a common platform for communication, coordination, and documentation. GitOps also enables code review, peer feedback, and knowledge sharing among team members.

- Enhanced security: GitOps enhances security by using Git as a secure and encrypted storage for configuration files. GitOps also enforces access control, authentication, authorization, and auditability for configuration changes.
- Increased efficiency: GitOps increases efficiency by using Git as a single source of truth for configuration management. GitOps also reduces manual errors, inconsistencies, and conflicts by automating configuration updates and deployments.
- Higher quality: GitOps improves quality by using Git as a reliable and reproducible record for configuration history. GitOps also ensures compliance with best practices, standards, and regulations by enforcing configuration policies.

According to a report by Red Hat, 83% of IT leaders surveyed said that they are either already using or planning to use GitOps within the next two years. The report also identifies some of the key drivers and challenges for GitOps adoption. Some of the drivers are:

- The increasing complexity and diversity of software development environments, which require more integration, automation, and governance.
- The rising demand for faster delivery and higher quality of software products and services, which require more visibility, measurement, and optimization.
- The growing awareness and maturity of GitOps practices and methodologies, which provide more guidance, support, and benefits.

Some of the challenges are:

- The lack of standardization and interoperability among different tools, platforms, and data sources that are used by the teams.
- The difficulty of changing the culture and mindset of the organization to embrace GitOps principles and values.
- The scarcity of skills and expertise in GitOps among the developers and operators.

Internal Developer Portals

Internal developer portals are web-based platforms that provide developers within an organization with access to the tools, resources, and information they need to build, test, and deploy software applications. Internal developer portals can include features such as documentation, tutorials, code samples, APIs, SDKs, libraries, frameworks, testing tools, deployment tools, and more. Internal developer portals are significant for software development teams, as they can help them improve their productivity, quality, and innovation.

Internal developer portals can improve collaboration among software developers and other stakeholders by facilitating communication, coordination, and knowledge sharing. Internal developer portals can enable developers to:

- Communicate with each other and with other teams or departments through forums, chats, blogs, or feedback mechanisms. This can help developers exchange ideas, opinions, or suggestions, as well as resolve issues or problems.

- Coordinate their actions and workflows through project management tools, version control systems, or issue trackers. This can help developers align their goals, tasks, and schedules, as well as monitor their progress and performance.
- Share their knowledge and expertise through documentation, tutorials, code samples, or best practices. This can help developers learn from each other and reuse existing solutions or components.

Internal developer portals are evolving with the changing needs and expectations of software developers and organizations. Some of the emerging trends in internal developer portals are:

- Personalization: Internal developer portals are becoming more personalized and customized to suit the preferences and profiles of individual developers or teams. This can help developers access the most relevant and useful information and tools for their specific needs or interests.
- Gamification: Internal developer portals are incorporating gamification elements such as badges, points, leaderboards, or rewards to motivate and engage developers. This can help developers improve their skills, performance, or contribution to the organization.
- Automation: Internal developer portals are leveraging automation technologies such as AI or ML to simplify or streamline the software development processes and tasks. This can help developers reduce errors, save time, or increase efficiency.

Open-source Program Office

Open source is a software development and distribution model that allows anyone to access, modify, and share the source code of software applications. Open source software is typically developed and maintained by a community of developers who collaborate and contribute to the project. Open source software can offer several benefits, such as lower costs, higher quality, faster innovation, and greater flexibility. Many corporations have recognized the value and potential of open source software and have adopted open source initiatives to leverage its advantages. Open source initiatives can include using, contributing to, or creating open source software for internal or external purposes.

An open-source program office is a dedicated team or department within a corporation that oversees and supports the open source initiatives of the organization. An open-source program office can help the organization achieve its open source goals and objectives, as well as address the challenges and risks associated with open source software. An open-source program office can provide various services and functions, such as:

- Developing and implementing an open source strategy and policy that aligns with the organization’s vision, mission, and values.
- Establishing and enforcing best practices and standards for open source software development, licensing, compliance, security, and governance.
- Educating and training the employees and stakeholders on the benefits, opportunities, and responsibilities of open source software.
- Managing and coordinating the open source projects and activities of the organization, such as selecting, evaluating, adopting, contributing to, or releasing open source software.
- Building and maintaining relationships with the open source community, such as participating in events, forums, or networks, providing feedback or support, or sponsoring or hosting open source projects.

Some examples of successful open-source program offices are:

- Google: Google is one of the leading technology companies that uses and contributes to a wide range of open source software projects, such as Android, Chrome, TensorFlow, Kubernetes, Angular, and more. Google has an open-source program office that manages its open source activities and promotes its open source culture. The Google open-source program office provides various resources and tools for Google employees and external developers who are involved in open source software.
- Microsoft: Microsoft is another leading technology company that has embraced open source software in recent years. Microsoft uses and contributes to many open source software projects, such as Linux, VS Code, .NET Core, React Native, TypeScript, and more. Microsoft also creates and supports several popular open source software projects, such as Azure DevOps, PowerShell Core, Windows Terminal, Visual Studio Code Extension API. Microsoft has an open-source program office that coordinates its open source efforts and fosters its open source community. The Microsoft open-source program office provides various guidelines and resources for Microsoft employees and external developers who are interested in open source software.
- Facebook: Facebook is another leading technology company that relies on and contributes to many open source software projects, such as React, PyTorch, GraphQL, Hack, PrestoDB. Facebook also develops and maintains several influential open source software projects, such as React Native, HHVM, RocksDB, Proxygen, Infer, Hermes, etc. Facebook has an open-source program office that oversees its open source activities and cultivates its open source ecosystem. The Facebook open-source program office provides various tools and platforms for Facebook employees and external developers who are engaged in open source software.

Open-source program offices are becoming more common and important in enterprises that want to benefit from or contribute to the open source movement. According to a report by The Linux Foundation and The TODO Group, 53% of the surveyed organizations have established an open-source program office or plan to create one soon. The report also identifies some of the key trends and challenges for open-source program offices. Some of the trends are:

- The increasing adoption of cloud computing technologies (such as Kubernetes) that are based on or compatible with open source software.
- The growing demand for digital transformation (such as AI/ML) that requires more integration (such as APIs), innovation (such as microservices), and collaboration (such as DevOps) among different software applications or systems.
- The rising popularity of inner-source (the use of open-source practices within an organization) that can improve the efficiency (such as code reuse), quality (such as code review), and culture (such as knowledge sharing) of software development teams.

Some of the challenges are:

- The lack of awareness or understanding of the value or potential of open-source software among senior management or business leaders.
- The difficulty of measuring or demonstrating the impact or return on investment (ROI) of open-source software initiatives or activities.

- The complexity or diversity of open-source software licensing, compliance, security, and governance issues or requirements.

Theme 3: Pervasive Cloud

Cloud Computing

Cloud computing is a computing paradigm that provides on-demand access to shared and scalable computing resources, such as servers, storage, networks, software, and applications, over the internet. Cloud computing has evolved over time to meet the changing needs and expectations of users and providers. According to the National Institute of Standards and Technology (NIST), cloud computing has five essential characteristics: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. Cloud computing also has three main service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Cloud computing also has four main deployment models: public cloud, private cloud, hybrid cloud, and community cloud.

Cloud computing offers numerous services and benefits for different users and domains, such as cost reduction, performance improvement, scalability enhancement, security assurance, and innovation enablement. Some of the key services and trends in cloud computing are:

- Artificial Intelligence (AI) and Machine Learning (ML): AI and ML are branches of computer science that deal with the creation and application of intelligent systems that can learn from data and perform tasks such as perception, recognition, prediction, and generation. Cloud computing can provide AI and ML services such as data storage, processing, analysis, modeling, training, inference, and deployment. Cloud computing can also leverage AI and ML techniques to optimize its own operations and performance.
- Internet of Things (IoT): IoT is a network of physical devices or objects that are embedded with sensors, actuators, or software that can collect, exchange, or act on data. Cloud computing can provide IoT services such as data management, device management, connectivity management, security management, and application development. Cloud computing can also enable IoT applications such as smart homes, smart cities, smart industries, smart healthcare, and smart agriculture.
- Edge Computing: Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location or source of data generation or consumption. Edge computing can complement cloud computing by providing low latency, high bandwidth, real-time processing, privacy preservation, and energy efficiency for time-sensitive or resource-constrained applications.

Cloud computing has a significant impact on various industries and sectors, such as business, education, healthcare, entertainment, government, and more. Cloud computing can help these industries and sectors achieve their goals and objectives by providing them with various solutions and opportunities. Some examples are:

- Business: Cloud computing can help businesses improve their productivity, efficiency, agility, competitiveness, and profitability by providing them with various solutions such as e-commerce platforms, customer relationship management systems, enterprise resource planning systems, business intelligence tools, and more.

- Education: Cloud computing can help education providers and learners improve their access, quality, affordability, flexibility, and effectiveness of education by providing them with various solutions such as online learning platforms, digital libraries, collaborative tools, assessment tools, and more.
- Healthcare: Cloud computing can help healthcare providers and patients improve their diagnosis, treatment, prevention, and research of health conditions by providing them with various solutions such as electronic health records, telemedicine platforms, medical imaging systems, drug discovery tools, and more.
- Entertainment: Cloud computing can help entertainment producers and consumers improve their creation, distribution, consumption of entertainment content by providing them with various solutions such as streaming platforms, gaming platforms, social media platforms, content creation tools, and more.
- Government: Cloud computing can help government agencies and citizens improve their governance service delivery by providing them with various solutions such as e-government platforms, digital identity systems, smart city systems, disaster management systems, and more.

Augmented FinOps

Augmented FinOps is a term that refers to the use of artificial intelligence (AI) and automation to enhance and support the FinOps practices and principles. FinOps is a cultural and operational model that aims to maximize the value and minimize the cost of cloud computing resources by aligning the financial, technical, and business goals of an organization. Augmented FinOps can help organizations achieve their FinOps objectives by providing them with various capabilities and benefits, such as:

- Data-driven decision making: Augmented FinOps can help organizations make better and faster decisions about their cloud spending and usage by providing them with data analysis, visualization, and reporting tools. Augmented FinOps can also leverage AI techniques such as machine learning (ML) and natural language processing (NLP) to generate insights, recommendations, and forecasts from the data.
- Process automation: Augmented FinOps can help organizations automate and streamline their FinOps processes and workflows by providing them with automation tools such as bots, scripts, or triggers. Augmented FinOps can also use AI techniques such as robotic process automation (RPA) and intelligent automation (IA) to perform tasks such as data collection, invoice processing, or anomaly detection.
- Continuous improvement: Augmented FinOps can help organizations improve their FinOps performance and maturity by providing them with feedback and guidance tools such as dashboards, alerts, or benchmarks. Augmented FinOps can also use AI techniques such as reinforcement learning (RL) and deep learning (DL) to learn from the data and optimize the outcomes.

Augmented FinOps can help organizations implement various cost optimization strategies for their cloud computing resources by providing them with tools and techniques that can help them:

- Plan and budget: Augmented FinOps can help organizations plan and budget their cloud spending and usage by providing them with tools such as cost calculators, estimators, or simulators. Augmented FinOps can also use AI techniques such as predictive analytics or scenario analysis to forecast the future demand and cost of cloud resources.

- Monitor and track: Augmented FinOps can help organizations monitor and track their cloud spending and usage by providing them with tools such as cost management platforms, accounting systems, or governance frameworks. Augmented FinOps can also use AI techniques such as anomaly detection or pattern recognition to identify and alert any deviations or issues in the cloud spending or usage.
- Optimize and save: Augmented FinOps can help organizations optimize and save on their cloud spending and usage by providing them with tools such as cost optimization platforms, pricing models, or policies. Augmented FinOps can also use AI techniques such as dynamic pricing or resource allocation to adjust the cloud spending or usage according to the demand or performance.

Augmented FinOps can help organizations achieve financial transparency for their cloud computing resources by providing them with tools and techniques that can help them:

- Measure and report: Augmented FinOps can help organizations measure and report their cloud spending and usage by providing them with tools such as metrics, indicators, or reports. Augmented FinOps can also use AI techniques such as natural language generation (NLG) or data visualization to present the data in a clear and understandable way.
- Share and communicate: Augmented FinOps can help organizations share and communicate their cloud spending and usage by providing them with tools such as collaboration platforms, notification systems, or feedback mechanisms. Augmented FinOps can also use AI techniques such as natural language understanding (NLU) or sentiment analysis to understand and respond to the queries or comments from the stakeholders.
- Align and govern: Augmented FinOps can help organizations align and govern their cloud spending and usage by providing them with tools such as frameworks, standards, or best practices. Augmented FinOps can also use AI techniques such as rule-based systems or decision support systems to enforce or recommend the appropriate actions or policies for the cloud spending or usage.

Cloud Development Environment

Cloud development environments are web-based platforms that provide developers with the tools and resources they need to build, test, and deploy cloud-native applications. Cloud-native applications are applications that are designed and optimized for the cloud computing paradigm, which offers on-demand access to shared and scalable computing resources over the internet. Cloud development environments enable cloud-native development by allowing developers to:

- Access and use cloud services and APIs, such as storage, databases, messaging, or machine learning, that can enhance the functionality and performance of their applications.
- Leverage cloud infrastructure and capabilities, such as servers, containers, or serverless functions, that can handle the scalability and reliability of their applications.
- Adopt cloud best practices and standards, such as microservices, DevOps, or CI/CD, that can improve the quality and efficiency of their development processes.

Cloud development environments offer various tools and platforms for different aspects of cloud-native development, such as coding, testing, debugging, deploying, or collaborating. Some of the tools and platforms are:

- Code editors and IDEs: These are tools that allow developers to write, edit, and run code for their applications. Some examples are CodeSandbox, StackBlitz, Repl.it, AWS Cloud9, Google Cloud Shell, or Visual Studio Code.
- Testing and debugging tools: These are tools that allow developers to test and debug their applications for errors, bugs, or defects. Some examples are Postman, Cypress, Selenium, Chrome DevTools, or Firefox Developer Tools.
- Deployment and hosting platforms: These are platforms that allow developers to deploy and host their applications on the cloud infrastructure. Some examples are Heroku, Netlify, Vercel, Firebase, or Azure Static Web Apps.
- Collaboration and communication tools: These are tools that allow developers to collaborate and communicate with other developers or stakeholders on their projects. Some examples are GitHub, GitLab, Bitbucket, Slack, or Discord.

Cloud development environments can improve developer productivity by providing them with various benefits and advantages, such as:

- Accessibility: Cloud development environments can be accessed from any device or location that has an internet connection. This allows developers to work from anywhere and anytime, without the need to install or update any software or hardware on their own devices or servers.
- Flexibility: Cloud development environments can be customized and configured to suit the preferences and needs of individual developers or teams. This allows developers to choose, combine, and modify different tools and platforms for their projects, as well as integrate them with other services or systems.
- Scalability: Cloud development environments can scale up or down according to the demand or performance of the applications. This allows developers to handle variable and unpredictable workloads or demands without worrying about the capacity or availability of the resources.

Cloud Sustainibility

Cloud sustainability is a term that refers to the environmental and social impact of cloud computing, which is a computing paradigm that provides on-demand access to shared and scalable computing resources over the internet. Cloud sustainability can be measured by various indicators, such as energy consumption, carbon footprint, water usage, waste generation, or social responsibility. Cloud sustainability can also be influenced by various factors, such as the location, design, operation, or optimization of the cloud infrastructure and services. Cloud sustainability is an important issue for cloud users and providers, as it can affect their costs, performance, reputation, and compliance.

Green data centers are data centers that are designed and operated to minimize their environmental impact and maximize their energy efficiency. Green data centers can adopt various strategies and technologies to achieve their goals, such as:

- Renewable energy sources: Green data centers can use renewable energy sources, such as solar, wind, or hydro power, to power their operations and reduce their reliance on fossil fuels. For example, Google claims that its data centers are powered by 100% renewable energy since 2017.
- Energy-efficient equipment: Green data centers can use energy-efficient equipment, such as servers, storage devices, or cooling systems, to reduce their energy consumption and heat generation. For example, Facebook uses custom-designed servers that are optimized for performance and power efficiency.

- Waste heat recovery: Green data centers can use waste heat recovery systems, such as heat pumps or heat exchangers, to capture and reuse the excess heat generated by their equipment for other purposes, such as heating or cooling buildings. For example, Amazon uses waste heat from its data center in Seattle to heat its nearby office buildings.

Corporate responsibility is a term that refers to the ethical and social obligations of corporations towards their stakeholders and society. Corporate responsibility can include various aspects, such as governance, compliance, transparency, accountability, diversity, inclusion, or philanthropy. Corporate responsibility can also relate to cloud sustainability, as cloud users and providers can demonstrate their commitment and contribution to environmental and social causes by adopting or supporting cloud sustainability initiatives or practices. Some examples are:

- Green certifications: Cloud users and providers can obtain green certifications or labels from reputable organizations or authorities that verify and validate their cloud sustainability performance or standards. For example, Microsoft has achieved several green certifications for its Azure cloud services, such as ISO 14001 (environmental management system), ISO 50001 (energy management system), or LEED (green building rating system).
- Carbon offsetting: Cloud users and providers can offset their carbon emissions from their cloud activities by investing in projects or programs that reduce or remove greenhouse gases from the atmosphere. For example, Apple has launched a $200 million fund to support carbon removal projects from forestry and other natural ecosystems.
- Social impact: Cloud users and providers can use their cloud resources or capabilities to support social impact initiatives or causes that address global challenges or improve the lives of people and communities. For example, IBM has created a cloud platform called Call for Code that connects developers with humanitarian experts to create solutions for disaster response and climate change.

Cloud Native

Cloud-native is a term that describes a software development and delivery approach that leverages the advantages of cloud computing, such as scalability, reliability, and flexibility. Cloud-native architecture is the design and implementation of software applications that are built for the cloud, using cloud services and technologies. Cloud-native architecture follows four main principles:

- Microservices: Cloud-native applications are composed of small and independent units of functionality, called microservices, that communicate with each other through APIs. Microservices can be deployed, updated, or scaled independently, without affecting the rest of the application.
- Containers: Cloud-native applications are packaged and run in lightweight and isolated environments, called containers, that provide the necessary resources and dependencies for the application. Containers can be easily moved and replicated across different cloud platforms or environments.
- Orchestration: Cloud-native applications are managed and coordinated by an orchestration tool, such as Kubernetes, that automates the deployment, scaling, and healing of the application. Orchestration can also provide service discovery, load balancing, networking, and security for the application.

- DevOps: Cloud-native applications are developed and delivered using DevOps practices and tools, such as continuous integration (CI), continuous delivery (CD), or continuous testing (CT), that enable faster feedback loops and shorter time to market. DevOps can also foster collaboration and innovation among developers and operators.

Cloud-native adoption is a growing trend across various industries and sectors, such as business, education, healthcare, entertainment, government, and more. Cloud-native adoption can provide various benefits and opportunities for these industries and sectors, such as:

- Cost reduction: Cloud-native applications can reduce the upfront and operational costs of software development and maintenance by eliminating the need to purchase, install, or manage hardware or software infrastructure. Cloud-native applications can also leverage the pay-as-you-go pricing model of cloud services, which allows users to pay only for what they use.
- Performance improvement: Cloud-native applications can improve the performance and quality of software products and services by leveraging the scalability and reliability of cloud resources. Cloud-native applications can also use cloud services and technologies, such as AI or ML, to enhance their functionality and value.
- Innovation enablement: Cloud-native applications can enable innovation and differentiation by allowing users to experiment with new ideas or features faster and easier. Cloud-native applications can also adapt to changing customer needs or market demands more quickly and effectively.

Some examples of cloud-native applications across industries are:

- Business: Netflix is a leading entertainment company that uses cloud-native architecture to stream movies and shows to millions of customers worldwide. Netflix uses microservices to handle different aspects of its service, such as recommendations, ratings, or payments. Netflix also uses containers to package and run its microservices on different cloud platforms or regions. Netflix also uses orchestration to manage and scale its microservices according to the demand or performance. Netflix also uses DevOps to develop and deliver new features or updates faster and more frequently.
- Education: Coursera is a leading online learning platform that uses cloud-native architecture to provide courses and degrees from top universities and institutions to millions of learners worldwide. Coursera uses microservices to handle different aspects of its platform, such as content delivery, assessment, or certification. Coursera also uses containers to package and run its microservices on different cloud platforms or environments. Coursera also uses orchestration to manage and coordinate its microservices according to the demand or performance. Coursera also uses DevOps to develop and deliver new courses or updates faster and more frequently.
- Healthcare: Cerner is a leading healthcare technology company that uses cloud-native architecture to provide health information systems and solutions to thousands of healthcare providers worldwide. Cerner uses microservices to handle different aspects of its systems, such as electronic health records (EHR), population health management (PHM), or revenue cycle management (RCM). Cerner also uses containers to package and run its microservices on different cloud platforms or environments. Cerner also uses orchestration to manage and scale its microservices according to the demand or performance. Cerner also uses DevOps to develop and deliver new solutions or updates faster and more frequently.

Cloud-native technologies are evolving with the changing needs and expectations of users and providers. Some of the emerging trends in cloud-native technologies are:

- Serverless Computing: Serverless computing is a cloud computing model that abstracts away the server management from the users and provides them with event-driven functions that execute on demand. Serverless computing can complement cloud-native architecture by providing more scalability, flexibility, and simplicity for cloud-native applications.
- Service Mesh: Service mesh is a network layer that provides communication, coordination, and control among the microservices of a cloud-native application. Service mesh can enhance cloud-native architecture by providing more visibility, security, reliability, and resilience for cloud-native applications.
- Hybrid Cloud: Hybrid cloud is a cloud computing model that combines public and private cloud resources and services. Hybrid cloud can enable cloud-native architecture by providing more choice, diversity, and interoperability for cloud-native applications.

Cloud-out to Edge

Cloud-out to edge is a term that describes the extension of cloud services and capabilities to the edge of the network, where data is generated or consumed. Cloud-out to edge can enable cloud users and providers to leverage the advantages of both cloud and edge computing, such as scalability, reliability, flexibility, low latency, high bandwidth, real-time processing, privacy preservation, and energy efficiency. Cloud-out to edge can be achieved by various methods, such as:

- Cloudlets: Cloudlets are small-scale cloud servers that are deployed near the edge devices or users. Cloudlets can provide cloud services and resources, such as storage, computation, or analytics, to the edge devices or users. Cloudlets can also offload or cache some of the data or tasks from the edge devices or users to reduce their workload or improve their performance.
- Edge nodes: Edge nodes are devices or systems that are connected to the cloud and can perform some of the cloud functions at the edge. Edge nodes can include routers, gateways, switches, or servers that can provide network connectivity, security, or management for the edge devices or users. Edge nodes can also include smart devices, such as cameras, sensors, or actuators that can collect, process, or act on data at the edge.
- Edge services: Edge services are cloud services that are delivered or accessed at the edge of the network. Edge services can include application services, such as streaming, gaming, or IoT; platform services, such as serverless, containers, or microservices; or infrastructure services, such as networking, storage, or computing.

Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location or source of data generation or consumption. Edge computing can provide various benefits and opportunities for different users and domains, such as:

- Autonomous vehicles: Edge computing can help autonomous vehicles improve their navigation, safety, and efficiency by providing them with low latency, high bandwidth, real-time processing, and privacy preservation for their data and tasks. For example, Tesla uses edge computing to enable its self-driving cars to process and analyze the data from their cameras and sensors locally and make decisions accordingly.
- Smart cities: Edge computing can help smart cities improve their traffic flow, energy efficiency, public safety, and environmental quality by providing them with low latency, high bandwidth, real-time processing, and privacy

preservation for their data and tasks. For example, Barcelona uses edge computing to enable its smart city solutions such as smart parking, smart lighting, smart waste management, or smart security.

- Healthcare: Edge computing can help healthcare providers and patients improve their diagnosis, treatment, prevention, and research of health conditions by providing them with low latency, high bandwidth, real-time processing, and privacy preservation for their data and tasks. For example, Philips uses edge computing to enable its healthcare solutions such as remote patient monitoring (RPM), telemedicine ™, medical imaging (MI), or wearable devices.

Edge computing is a growing trend in the cloud computing industry as more users and providers recognize its value and potential. According to a report by Gartner, the global market for edge computing is expected to reach $250.6 billion by 2024 with a compound annual growth rate of 21%. The report also identifies some of the key drivers and challenges for edge computing adoption. Some of the drivers are:

- The increasing demand for digital transformation (such as AI/ML) that requires more integration (such as APIs), innovation (such as microservices), and collaboration (such as DevOps) among different software applications or systems.
- The rising popularity of IoT (Internet of Things) devices and applications that generate or consume large amounts of data at the edge of the network.
- The growing awareness and maturity of edge computing technologies and solutions that provide more guidance support and benefits.

Some of the challenges are:

- The complexity and diversity of edge computing environments and architectures that require more integration coordination and governance.
- The difficulty of changing the culture and mindset of the organization to embrace edge computing principles and values.
- The scarcity of skills and expertise in edge computing among the developers and operators.

WebAssembly

WebAssembly is a binary instruction format that can run on any web browser or platform. WebAssembly can enable web developers to write high-performance and portable code using languages such as C, C++, Rust, or Go, and compile it to WebAssembly modules that can be executed on the web. WebAssembly can also play a significant role in the cloud computing paradigm, which provides on-demand access to shared and scalable computing resources over the internet. WebAssembly can enable cloud developers to:

- Deploy and run cloud applications on any cloud platform or environment, without the need to install or configure any software or hardware dependencies. WebAssembly can also leverage the cloud infrastructure and capabilities, such as servers, containers, or serverless functions, to handle the scalability and reliability of cloud applications.
- Integrate and interoperate with different cloud services and technologies, such as storage, databases, messaging, or machine learning, that can enhance the functionality and performance of cloud applications. WebAssembly can also use cloud APIs and SDKs to communicate and interact with other cloud applications or systems.

- Adopt and follow cloud best practices and standards, such as microservices, DevOps, or CI/CD, that can improve the quality and efficiency of cloud development processes and outcomes. WebAssembly can also foster collaboration and innovation among cloud developers and operators.

WebAssembly has various use cases and advantages for different users and domains, such as:

- Gaming: WebAssembly can help game developers create and deliver immersive and interactive games that can run on any web browser or device, without the need to download or install any plugins or software. WebAssembly can also provide game developers with high performance, low latency, real-time processing, and graphics rendering for their games. For example, Unity is a game engine that uses WebAssembly to enable game developers to build and deploy web-based games.
- Media: WebAssembly can help media producers and consumers create and consume high-quality and engaging media content, such as images, videos, or audio, that can run on any web browser or device, without the need to download or install any codecs or software. WebAssembly can also provide media producers and consumers with high performance, low bandwidth, real-time processing, and compression for their media content. For example, Squoosh is a web-based image compression tool that uses WebAssembly to enable users to compress their images faster and easier.
- Education: WebAssembly can help education providers and learners improve their access, quality, affordability, flexibility, and effectiveness of education by providing them with web-based learning platforms, tools, or resources that can run on any web browser or device, without the need to download or install any software or hardware. WebAssembly can also provide education providers and learners with high performance, low latency, real-time processing, and interactivity for their learning content. For example, WasmBoy is a web-based emulator that uses WebAssembly to enable users to play Game Boy games on their web browsers.

WebAssembly is a promising technology that has the potential to transform the way web and cloud applications are developed and delivered. WebAssembly is evolving with the changing needs and expectations of developers and users. Some of the emerging trends in WebAssembly are:

- Multi-language support: WebAssembly is expanding its support for more programming languages that can be compiled to WebAssembly modules. This can enable developers to use their preferred or familiar languages for web or cloud development. For example, Blazor is a framework that uses WebAssembly to enable developers to use C# for web development.
- Interoperability: WebAssembly is improving its interoperability with other web technologies such as JavaScript (JS), HTML (HyperText Markup Language), or CSS (Cascading Style Sheets). This can enable developers to integrate or reuse existing web components or libraries for their web or cloud applications. For example, Emscripten is a tool that uses WebAssembly to enable developers to port existing C/C++ code or libraries to the web.
- Standardization: WebAssembly is becoming more standardized and widely adopted by various organizations or authorities that regulate or promote web or cloud technologies. This can enable developers to follow common guidelines or specifications for their web or cloud applications. For example, W3C (World Wide Web Consortium) is an organization that develops standards for the web. W3C has established a working group for WebAssembly that defines its core specification and extensions.

Theme 4: Human-centric Security and Privacy

AI Security, Risk, and Management

AI Trism (Trustworthy AI) is a term that refers to the development and deployment of artificial intelligence (AI) systems that are trustworthy, reliable, and beneficial for humans and society. AI Trism can help build trust in AI systems by ensuring that they meet the following criteria:

- Transparency: AI systems should be transparent and explainable, meaning that they should provide clear and understandable information about their goals, capabilities, limitations, and outcomes. Transparency can help users and stakeholders understand how and why AI systems make decisions or actions, as well as their potential impacts or risks.
- Responsibility: AI systems should be responsible and accountable, meaning that they should follow ethical principles and values, as well as comply with legal and regulatory frameworks. Responsibility can help users and stakeholders ensure that AI systems respect human dignity, rights, and interests, as well as prevent or mitigate any harm or damage.
- Security: AI systems should be secure and robust, meaning that they should protect the data and information they use or produce, as well as perform reliably and consistently under various conditions. Security can help users and stakeholders prevent or defend against any threats or attacks on AI systems, such as hacking, tampering, or manipulation.

Ethical AI development is the process of designing and creating AI systems that adhere to the ethical principles and values of humans and society. Ethical AI development can help achieve AI Trism by involving various methods and practices, such as:

- Stakeholder engagement: Ethical AI development should engage with different stakeholders, such as users, customers, developers, operators, regulators, or experts, who are affected by or interested in AI systems. Stakeholder engagement can help identify and address the needs, expectations, and concerns of different parties, as well as foster collaboration and communication among them.
- Value alignment: Ethical AI development should align with the values and norms of humans and society, such as fairness, justice, privacy, or safety. Value alignment can help ensure that AI systems respect and promote the common good and well-being of humans and society, as well as avoid or minimize any negative or adverse outcomes.
- Impact assessment: Ethical AI development should assess the impact of AI systems on humans and society, both before and after their deployment. Impact assessment can help evaluate the benefits and risks of AI systems, as well as monitor and measure their performance and effects.

Trustworthiness standards are the guidelines or specifications that define the requirements or criteria for achieving trustworthy AI systems. Trustworthiness standards can help support AI Trism by providing various tools and resources, such as:

- Frameworks: Trustworthiness frameworks are the conceptual models or structures that describe the key elements or dimensions of trustworthy AI systems. Trustworthiness frameworks can help users and stakeholders understand the scope and complexity of trustworthy AI systems, as well as their interrelationships and dependencies.
- Principles: Trustworthiness principles are the general rules or recommendations that guide the development and deployment of trustworthy AI systems. Trustworthiness principles can help users and stakeholders establish the goals and objectives of trustworthy AI systems, as well as their ethical implications.
- Metrics: Trustworthiness metrics are the quantitative or qualitative measures that evaluate the performance or quality of trustworthy AI systems. Trustworthiness metrics can help users and stakeholders verify and validate the compliance or achievement of trustworthy AI systems, as well as their improvement or optimization.

Some examples of trustworthiness standards are:

- IEEE (Institute of Electrical and Electronics Engineers) is an organization that develops standards for engineering and technology. IEEE has launched a project called P7000 that aims to create standards for ethical aspects of trustworthy AI systems.
- ISO (International Organization for Standardization) is an organization that develops standards for various domains and industries. ISO has established a committee called ISO/IEC JTC 1/SC 42 that aims to create standards for technical aspects of trustworthy AI systems.
- EU (European Union) is a political and economic union of European countries. EU has published a document called Ethics Guidelines for Trustworthy Artificial Intelligence that aims to provide a framework for ethical aspects of trustworthy AI systems.

Cybersecurity Mesh Architecture

CSMA (Cybersecurity Mesh Architecture) is a term that describes a distributed and modular approach to cybersecurity that allows users to access and protect any digital asset, regardless of its location or network. CSMA is based on the idea of shifting the security perimeter from the network to the individual, meaning that each user or device can define and enforce its own security policies and rules. CSMA can enable users to create and manage a flexible and personalized security mesh that covers their data, applications, devices, or identities.

CSMA can enhance data protection by providing various benefits and advantages, such as:

- Scalability: CSMA can scale up or down according to the demand or performance of the security mesh. CSMA can also leverage cloud services and technologies, such as servers, containers, or serverless functions, to handle the scalability and reliability of the security mesh.
- Flexibility: CSMA can offer more flexibility and customization options for users by allowing them to choose, combine, and modify different security components or services for their security mesh. CSMA can also integrate with different security standards or frameworks, such as Zero Trust, SASE (Secure Access Service Edge), or NIST (National Institute of Standards and Technology).
- Visibility: CSMA can provide more visibility and control for users by allowing them to monitor and manage their security mesh from a single dashboard or platform. CSMA can also use AI (Artificial Intelligence) or ML (Machine Learning) techniques to generate insights, alerts, or recommendations from the security data.

CSMA has various industry applications for different sectors and domains, such as business, healthcare, education, government, and more. CSMA can help these sectors and domains achieve their security goals and objectives by providing them with various solutions and opportunities. Some examples are:

- Business: CSMA can help businesses improve their productivity, efficiency, agility, competitiveness, and profitability by providing them with various solutions such as remote work platforms, e-commerce platforms, customer relationship management systems, business intelligence tools, and more. CSMA can also help businesses protect their data, assets, customers, and reputation from cyber threats or attacks.
- Healthcare: CSMA can help healthcare providers and patients improve their diagnosis, treatment, prevention, and research of health conditions by providing them with various solutions such as electronic health records (EHR), telemedicine platforms, medical imaging systems, drug discovery tools, wearable devices, and more. CSMA can also help healthcare providers and patients protect their data, privacy, safety, and well-being from cyber threats or attacks.
- Education: CSMA can help education providers and learners improve their access, quality, affordability, flexibility, and effectiveness of education by providing them with various solutions such as online learning platforms, digital libraries, collaborative tools, assessment tools, certification tools, and more. CSMA can also help education providers and learners protect their data, privacy, security, and integrity from cyber threats or attacks.

Generative Cybersecurity AI

Generative AI is a branch of artificial intelligence (AI) that deals with the creation and synthesis of new data or content, such as images, text, audio, or video. Generative AI can use various techniques and models, such as generative adversarial networks (GANs), variational autoencoders (VAEs), or transformers, to learn from existing data and generate realistic and diverse outputs. Generative AI can play a significant role in cybersecurity, which is the protection of data and systems from cyber threats or attacks. Generative AI can enable cybersecurity by providing various capabilities and benefits, such as:

- Data augmentation: Generative AI can help cybersecurity by augmenting the data that is used for training or testing cybersecurity models or systems. Generative AI can create synthetic or modified data that can increase the quantity, quality, or diversity of the data, as well as address the issues of data scarcity, imbalance, or privacy.
- Adversarial defense: Generative AI can help cybersecurity by defending against adversarial attacks that aim to fool or evade cybersecurity models or systems. Generative AI can detect or mitigate adversarial examples that are maliciously crafted to manipulate the input or output of the models or systems, such as images, text, audio, or video.
- Threat emulation: Generative AI can help cybersecurity by emulating the behavior or tactics of cyber attackers that aim to compromise or damage cybersecurity models or systems. Generative AI can generate realistic and diverse scenarios or simulations that can test or improve the resilience or performance of the models or systems.

Generative AI can help cybersecurity by enhancing the threat detection and response capabilities of cybersecurity models or systems. Threat detection and response are the processes of identifying and responding to cyber threats or attacks that

target data or systems. Generative AI can enable threat detection and response by providing various solutions and opportunities, such as:

- Anomaly detection: Generative AI can help threat detection by detecting anomalies or deviations in the data or systems that indicate potential cyber threats or attacks. Generative AI can use techniques such as GANs or VAEs to model the normal behavior or state of the data or systems, and compare it with the observed behavior or state to identify any anomalies.
- Incident response: Generative AI can help threat response by responding to incidents or events that result from cyber threats or attacks. Generative AI can use techniques such as transformers or reinforcement learning (RL) to generate actions or recommendations that can mitigate or resolve the incidents, such as alerts, reports, patches, or countermeasures.
- Forensic analysis: Generative AI can help threat response by analyzing the evidence or traces that are left behind by cyber threats or attacks. Generative AI can use techniques such as transformers or natural language generation (NLG) to generate summaries or explanations that can provide insights into the nature, origin, impact, or attribution of the threats or attacks.

Generative cybersecurity is a growing and evolving field that has the potential to transform the way cybersecurity is practiced and performed. Generative cybersecurity is advancing with the changing needs and expectations of users and providers. Some of the emerging trends in generative cybersecurity are:

- Multi-modal generation: Generative cybersecurity is expanding its scope and capability by generating multi-modal data or content that involves multiple types or formats, such as images, text, audio, video, etc. Multi-modal generation can enable generative cybersecurity to handle more complex and diverse cyber threats or attacks that involve multiple modes of communication or interaction.
- Explainable generation: Generative cybersecurity is improving its transparency and accountability by generating explainable data or content that provides clear and understandable information about its sources, methods, outcomes, etc. Explainable generation can enable generative cybersecurity to build trust and confidence among users and stakeholders who rely on its outputs.
- Collaborative generation: Generative cybersecurity is enhancing its efficiency and effectiveness by generating collaborative data or content that involves multiple agents or parties who cooperate or compete with each other. Collaborative generation can enable generative cybersecurity to leverage the collective intelligence and creativity of humans and machines who work together to achieve common goals.

Homomorphic Encryption

Homomorphic encryption is a type of encryption that allows computations to be performed on encrypted data without decrypting it first. Homomorphic encryption can preserve the privacy and security of the data while enabling useful operations and analysis on it. Homomorphic encryption can be classified into three categories, depending on the complexity and functionality of the computations:

- Partially homomorphic encryption (PHE): PHE supports only one type of operation (either addition or multiplication) on encrypted data, but it can perform it an unlimited number of times. For example, Paillier cryptosystem is a PHE scheme that supports addition on encrypted data.

- Somewhat homomorphic encryption (SHE): SHE supports both addition and multiplication on encrypted data, but it can perform them only a limited number of times, depending on the noise or error that accumulates during the computations. For example, Brakerski-Gentry-Vaikuntanathan (BGV) scheme is a SHE scheme that supports both addition and multiplication on encrypted data.
- Fully homomorphic encryption (FHE): FHE supports both addition and multiplication on encrypted data, and it can perform them an unlimited number of times, without any noise or error. FHE can also support any arbitrary function or algorithm on encrypted data, as any function or algorithm can be expressed as a combination of addition and multiplication. For example, Gentry's scheme is the first FHE scheme that supports any arbitrary function on encrypted data.

Homomorphic encryption can enable privacy-preserving data processing by allowing users to outsource their data and computations to untrusted third parties, such as cloud servers or service providers, without revealing their data or compromising their results. Homomorphic encryption can provide various benefits and opportunities for privacy-preserving data processing, such as:

- Data confidentiality: Homomorphic encryption can protect the data from unauthorized access or disclosure by encrypting it before sending it to the third parties. The third parties can only see the encrypted data, which is meaningless and useless for them, and they cannot decrypt it without the secret key that only the user has.
- Data integrity: Homomorphic encryption can ensure the data from unauthorized modification or corruption by verifying the correctness and validity of the computations on the encrypted data. The user can check if the computations are performed correctly and accurately by decrypting the results and comparing them with the expected outcomes.
- Data utility: Homomorphic encryption can maintain the data from losing its value or usefulness by enabling various operations and analysis on the encrypted data. The user can benefit from the services or functions that the third parties provide, such as storage, computation, analytics, or machine learning, without sacrificing their data privacy or security.

Homomorphic encryption has various practical applications for different domains and sectors, such as business, healthcare, education, government, and more. Homomorphic encryption can help these domains and sectors achieve their goals and objectives by providing them with various solutions and opportunities. Some examples are:

- Business: Homomorphic encryption can help businesses improve their productivity, efficiency, agility, competitiveness, and profitability by providing them with various solutions such as cloud computing platforms, customer relationship management systems, business intelligence tools, and more. Homomorphic encryption can also help businesses protect their data, assets, customers, and reputation from cyber threats or attacks.
- Healthcare: Homomorphic encryption can help healthcare providers and patients improve their diagnosis, treatment, prevention, and research of health conditions by providing them with various solutions such as electronic health records (EHR), telemedicine platforms, medical imaging systems, drug discovery tools, wearable devices, and more. Homomorphic encryption can also help healthcare providers and patients protect their data, privacy, safety, and well-being from cyber threats or attacks.

- Education: Homomorphic encryption can help education providers and learners improve their access, quality, affordability, flexibility, and effectiveness of education by providing them with various solutions such as online learning platforms, digital libraries, collaborative tools, assessment tools, certification tools, and more. Homomorphic encryption can also help education providers and learners protect their data, privacy, security, and integrity from cyber threats or attacks.

Post-Quantum Cryptography

Quantum computing is a computing paradigm that uses quantum physics to perform computations that are faster and more powerful than classical computers. Quantum computing can offer various benefits and opportunities for different domains and sectors, such as artificial intelligence, cryptography, or optimization. However, quantum computing can also pose serious threats and challenges for the security and privacy of data and systems, especially for the current cryptographic schemes that are widely used to protect them. Quantum computing can potentially break or weaken the cryptographic schemes that are based on the hardness of mathematical problems, such as factoring large numbers or finding discrete logarithms. These problems are easy to solve for quantum computers, but hard to solve for classical computers. For example, Shor's algorithm is a quantum algorithm that can efficiently factor large numbers, and thus break the security of RSA encryption scheme. Therefore, there is an urgent need to prepare for the quantum computing threats and develop new cryptographic schemes that can resist or withstand the quantum attacks.

Post-quantum cryptography (PQC) is a branch of cryptography that deals with the design and analysis of cryptographic schemes that are secure against quantum attacks. PQC aims to provide security and privacy for data and systems in the post-quantum era, when quantum computers become widely available and accessible. PQC can be classified into two categories, depending on the approach and methodology of the cryptographic schemes:

- Quantum-resistant cryptography: Quantum-resistant cryptography is a type of PQC that uses classical (non-quantum) techniques and algorithms to create cryptographic schemes that are hard to break or solve by both classical and quantum computers. Quantum-resistant cryptography relies on the hardness of mathematical problems that are not affected by quantum algorithms, such as lattice problems, code problems, or multivariate problems. For example, NTRU is a quantum-resistant encryption scheme that is based on the hardness of finding the shortest vector in a lattice.
- Quantum cryptography: Quantum cryptography is a type of PQC that uses quantum (non-classical) techniques and algorithms to create cryptographic schemes that are based on the laws of quantum physics, such as superposition, entanglement, or uncertainty. Quantum cryptography can provide provable security and privacy for data and systems by exploiting the unique properties and behaviors of quantum particles, such as photons or electrons. For example, quantum key distribution (QKD) is a quantum cryptographic scheme that uses quantum communication to securely exchange secret keys between two parties.

PQC is a promising and evolving field that has the potential to transform the way data and systems are protected and secured in the quantum world. PQC is advancing with the changing needs and expectations of users and providers. Some of the emerging trends and challenges in PQC are:

- Standardization: PQC is becoming more standardized and widely adopted by various organizations or authorities that regulate or promote cybersecurity technologies and solutions. For example, NIST (National Institute of Standards and Technology) is an organization that develops standards for various domains and industries. NIST has launched a project called Post-Quantum Cryptography Standardization that aims to select and recommend PQC schemes for public-key encryption, key exchange, and digital signatures.
- Implementation: PQC is becoming more practical and feasible by improving its implementation and deployment on different platforms or devices. For example, Microsoft is a leading technology company that has implemented and deployed PQC schemes on its cloud services and products, such as Azure or Office 365. Microsoft has also created a toolkit called PQCrypto-VPN that allows users to test and experiment with PQC schemes on their virtual private networks (VPNs).
- Innovation: PQC is becoming more innovative and diverse by exploring new techniques or models for creating cryptographic schemes that can provide more security or functionality for data and systems. For example, homomorphic encryption is a type of encryption that allows computations to be performed on encrypted data without decrypting it first. Homomorphic encryption can enable privacy-preserving data processing by allowing users to outsource their data and computations to untrusted third parties without revealing their data or compromising their results.

Key Takeaways

The 2023 Hype Cycle is a report by Gartner that analyzes and evaluates the maturity and adoption of various emerging technologies and trends that have the potential to transform the world. The 2023 Hype Cycle identifies five stages of the technology life cycle: innovation trigger, peak of inflated expectations, trough of disillusionment, slope of enlightenment, and plateau of productivity. The 2023 Hype Cycle also categorizes the technologies and trends into six themes: cloud, AI, cybersecurity, edge, quantum, and generative. Some of the key takeaways from the 2023 Hype Cycle are:

- Cloud computing is the dominant and pervasive theme that underpins and enables most of the other technologies and trends. Cloud computing provides on-demand access to shared and scalable computing resources over the internet, such as servers, storage, networks, software, and applications. Cloud computing also offers various benefits and opportunities for different users and domains, such as cost reduction, performance improvement, scalability enhancement, security assurance, and innovation enablement.
- AI is the most disruptive and influential theme that drives and shapes most of the other technologies and trends. AI is a branch of computer science that deals with the creation and application of intelligent systems that can learn from data and perform tasks such as perception, recognition, prediction, and generation. AI also provides various benefits and opportunities for different users and domains, such as data analysis, automation, optimization, personalization, and creativity.
- Cybersecurity is the most critical and challenging theme that protects and secures most of the other technologies and trends. Cybersecurity is the protection of data and systems from cyber threats or attacks that target their confidentiality, integrity, or availability. Cybersecurity also provides various benefits and opportunities for different users and domains, such as data privacy, system reliability, threat mitigation, risk management, and compliance.

- Edge computing is the most emerging and promising theme that complements and extends most of the other technologies and trends. Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location or source of data generation or consumption. Edge computing also provides various benefits and opportunities for different users and domains, such as low latency, high bandwidth, real-time processing, privacy preservation, and energy efficiency.
- Quantum computing is the most advanced and futuristic theme that challenges and surpasses most of the other technologies and trends. Quantum computing is a computing paradigm that uses quantum physics to perform computations that are faster and more powerful than classical computers. Quantum computing also provides various benefits and opportunities for different users and domains, such as complex problem solving, cryptography breaking or enhancing, simulation modeling, or quantum communication.
- Generative technologies are the most creative and diverse theme that creates and synthesizes most of the other technologies and trends. Generative technologies are technologies that use AI or other techniques to create new data or content, such as images, text, audio, or video. Generative technologies also provide various benefits and opportunities for different users and domains, such as data augmentation, adversarial defense, threat emulation, media production or consumption.

Software Engineering

Software Engineering is an engineering branch that deals with the design, development, testing, and maintenance of software products. Software products are applications or systems that run on computers or other devices and perform specific tasks or functions. Software Engineering applies scientific and engineering principles to create reliable, efficient, and secure software products that meet the needs and expectations of users and customers.

Software Engineering involves various activities and processes that aim to ensure the quality and success of software products. Some of these activities and processes are:

- Software Requirements: This is the process of gathering, analyzing, and documenting the needs and objectives of the software product from the perspective of the users and customers. This helps to define the scope, features, functionalities, and constraints of the software product.
- Software Design: This is the process of creating a plan or blueprint for the software product that describes its architecture, components, interfaces, data structures, algorithms, and other aspects. This helps to ensure that the software product is feasible, consistent, modular, and maintainable.
- Software Implementation: This is the process of writing, testing, and debugging the code that implements the software design. This helps to ensure that the software product is functional, correct, and efficient.
- Software Testing: This is the process of verifying and validating that the software product meets the requirements and design specifications. This helps to ensure that the software product is reliable, secure, and free of errors or defects.
- Software Maintenance: This is the process of modifying and updating the software product after its delivery to fix bugs, improve performance, add new features, or adapt to changing environments. This helps to ensure that the software product remains useful and relevant over time.

Software Engineering faces many challenges and difficulties due to the complexity, diversity, and dynamism of software products. Some of these challenges are:

- Shifting Requirements: Software requirements may change frequently or unpredictably due to changing user needs, market demands, or business goals. This can affect the scope, cost, schedule, and quality of the software product.
- Technological Changes: Software technologies evolve rapidly and continuously, introducing new tools, frameworks, languages, platforms, standards, and paradigms. This can affect the compatibility, performance, security, and maintainability of the software product.
- Integration Issues: Software products often need to interact or integrate with other software products or systems that may have different architectures, protocols, formats, or functionalities. This can affect the interoperability, functionality, and reliability of the software product.
- Quality Assurance: Software products need to meet high standards of quality in terms of functionality, reliability, usability, efficiency, security, and maintainability. This requires rigorous testing, verification, validation, debugging, and documentation of the software product.
- Project Management: Software projects need to be planned, organized, coordinated, monitored, controlled, and delivered within budget, time frame ,and quality expectations. This requires effective communication ,collaboration ,and leadership among various stakeholders ,such as developers ,testers ,customers ,users ,and managers.

Software Engineering is a challenging but rewarding field that requires creativity, problem-solving, analytical thinking, and technical skills. Software engineers can create innovative solutions that can benefit various domains, such as education, healthcare, entertainment, business, and social good. Software Engineering is also a dynamic and evolving field that offers many opportunities for learning, growth, and career advancement.

REFERENCES

OpenAI, “ChatGPT,” 2022. [Online]. Available: https://openai.com/chatgpt. [Accessed: 05-Sep-2023].

A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, M. Chen et al., “Competitive programming with AlphaCode,” Science, vol. 374, no. 6576, pp. 1448–1454, Dec. 2021.

IBM Research, “AI-enriched Simulation,” 2020. [Online]. Available: https://research.ibm.com/science/ai-enriched-simulation/. [Accessed: 05-Sep-2023].

L. Ramos, A. Mullen, and P. den Hamer, “Innovation Insight: AI Simulation,” Gartner Research, Jan. 2023.

D. Matovski, “What is ‘Causal AI’ and why will it become increasingly important?” World Economic Forum, Jun. 23, 2023. [Online]. Available: https://www.weforum.org/agenda/2023/06/what-is-causal-ai-why-important/. [Accessed: 05-Sep-2023].

J. M. “Causal AI: rewriting the rules of artificial intelligence,” CyberNews, Jul. 10, 2023. [Online]. Available: https://cybernews.com/editorial/causal-ai-explained/. [Accessed: 05-Sep-2023].

D. Ratzesberger, “What is causal AI? Deterministic AI,” Dynatrace Blog, Jul. 25, 2023. [Online]. Available: https://www.dynatrace.com/news/blog/what-is-causal-ai-deterministic-ai/. [Accessed: 05-Sep-2023].

causaLens, “causaLens | a causal AI software company | Decision-making AI,” 2023. [Online]. Available: https://causalens.com/. [Accessed: 05-Sep-2023].

Senslytics, “Senslytics | Discover Causality,” 2023. [Online]. Available: https://www.senslytics.com/how-it-works. [Accessed: 05-Sep-2023].

Xplain Data, “Xplain Data | Discover Causality,” 2023. [Online]. Available: https://xplain-data.de/xplain-data-gartner-ai-hype-cycle-2023-causal-ai/. [Accessed: 05-Sep-2023].

M. Roehm, “Top 7 Open-Source Frameworks for Federated Learning,” Apheris, 9 September 2022. [Online]. Available: https://www.apheris.com/resources/blog/top-7-open-source-frameworks-for-federated-learning. [Accessed 5 September 2023].

Q. Yang, Y. Liu and T. Chen, Eds., Federated Learning: A Comprehensive Overview of Methods and Applications, Cham: Springer International Publishing, 2023.

Perri, Lori. , “What’s New in the 2023 Gartner Hype Cycle for Emerging Technologies,” Gartner, August 23, 2023. [online]. Available: https://www.gartner.com/en/articles/what-s-new-in-the-2023-gartner-hype-cycle-for-emerging-technologies (accessed Sep. 5, 2023).

S. Robinson, “What is Graph Data Science? Essential Overview (6 Minute Article),” Graphable Blog, Apr. 12, 2022. [Online]. Available: https://www.graphable.ai/blog/graph-data-science/. [Accessed: Sep. 05, 2023].

Neo4j, “Neo4j Graph Data Science | Graph Algorithms and ML | Graph Analytics,” 2023. [Online]. Available: https://neo4j.com/product/graph-data-science/. [Accessed: Sep. 05, 2023].

Neo4j, “Neo4j Graph Data Science - Developer Guides,” 2023. [Online]. Available: https://neo4j.com/developer/graph-data-science/ [Accessed: Sep. 05, 2023].

George Anadiotis, “Graph data science: What you need to know,” VentureBeat, Apr. 12, 2022. [Online]. Available: https://venturebeat.com/business/graph-data-science-what-you-need-to-know/. [Accessed: Sep. 05, 2023].

“Reinforcement learning - GeeksforGeeks,” 2023. [Online]. Available: https://www.geeksforgeeks.org/what-is-reinforcement-learning/. [Accessed: Sep. 05, 2023].

“What is Reinforcement Learning? – Overview of How it Works - Synopsys,” 2023. [Online]. Available: https://www.synopsys.com/ai/what-is-reinforcement-learning.html. [Accessed: Sep. 05, 2023].

“Value Stream Management Platforms Reviews and Ratings - Gartner,” 2023. [Online]. Available: https://www.gartner.com/reviews/market/value-stream-management-platforms. [Accessed: Sep. 05, 2023].

“Value Stream Management Solution | Atlassian VSM | Atlassian,” 2023. [Online]. Available: https://www.atlassian.com/solutions/value-stream-management. [Accessed: Sep. 05, 2023].

M. Bhat, “Gartner Publishes Detailed Market Guides for Value Stream Platforms,” Nov. 12, 2021. [Online]. Available: https://blogs.gartner.com/manjunath-bhat/2021/11/12/gartner-publishes-detailed-market-guides-for-value-stream-platforms/. [Accessed: Sep. 05, 2023].

"How to Use Value Stream Management Platforms for DevOps," Code Climate Blog, Aug. 25, 2021. [Online]. Available: https://codeclimate.com/blog/what-is-value-stream-management. [Accessed: Sep. 05, 2023].

"Value Stream Management Platform | The Ultimate Guide," Medium, Aug. 02, 2021. [Online]. Available: https://medium.com/digital-transformation-and-platform-engineering/value-stream-management-platform-the-ultimate-guide-4afd91d568b7. [Accessed: Sep. 05, 2023].

"AI-Augmented Software Engineering | Splunk," 2023. [Online]. Available: https://www.splunk.com/en_us/blog/learn/ai-augmented-software-engineering.html. [Accessed: Sep. 05, 2023].

Gartner, "Emerging Technologies: Critical Insights Into AI-Augmented Software Development," Dec. 18, 2020. [Online]. Available: https://www.gartner.com/en/documents/3994660. [Accessed: Sep. 05, 2023].

vFunction, "How AI-Augmented Software Engineering will Impact Our Businesses?," Mar. 28, 2023. [Online]. Available: https://vfunction.com/blog/how-ai-augmented-software-engineering-will-impact-business/. [Accessed: Sep. 05, 2023].

Postman, "API-Centric SaaS: The Future of Software Delivery," Dec. 15, 2020. [Online]. Available: 4. [Accessed: Sep. 05, 2023].

B. Dayley, "Accelerate Digital Transformation With an API-Centric (Headless) Architecture for Enterprise Applications," Gartner, Stamford, CT, USA, 2021. [Online]. Available: https://www.gartner.com/en/documents/3995739. [Accessed: Sep. 05, 2023].

M. Dvorak, "Rising Cloud Technologies: API-Centric SaaS, Cloudlets and Blockchain PaaS," Martin Dvorak Consulting, Jul. 10, 2021. [Online]. Available: http://martindvorak.com/rising-cloud-technologies-api-centric-saas-cloudlets-and-blockchain-paas/. [Accessed: Sep. 05, 2023].

A. Abhishek, "API Centric SaaS That Customers and Developers Will Love: Strategies for Design, Adoption and Monetization," Alok Abhishek's Blog, Jul. 16, 2023. [Online]. Available: https://alokabhishek.com/product-management/api-centric-saas-that-customers-and-developers-will-love-strategies-for-design-adoption-and-monetization/. [Accessed: Sep. 05, 2023].

D. Ziolkowski, "Internal Developer Portal: What it Is and Why You Need One," DevOps.com, May 5, 2022. [Online]. Available: https://devops.com/internal-developer-portal-what-it-is-and-why-you-need-one/. [Accessed: Sep. 05, 2023].

"Guide to Internal Developer Portals | Port," Port, 2021. [Online]. Available: https://www.getport.io/blog/guide-to-internal-developer-portals. [Accessed: Sep. 05, 2023].

"The Four Pillars of Internal Developer Portals | Port," Port, Aug. 9, 2023. [Online]. Available: https://www.getport.io/blog/the-four-pillars-of-internal-developer-portals. [Accessed: Sep. 05, 2023].

"Internal Developer Portal: What It Is and Why You Need One," The New Stack, Jan. 24, 2023. [Online]. Available: https://thenewstack.io/internal-developer-portal-what-it-is-and-why-you-need-one/. [Accessed: Sep. 05, 2023].

"Open Source Programs Office (OSPO) - Johns Hopkins University," Johns Hopkins University, 2021. [Online]. Available: https://drcc.library.jhu.edu/open-source-programs-office/. [Accessed: Sep. 05, 2023].

“Creating an Open Source Program - Linux Foundation,” Linux Foundation, 2017. [Online]. Available: https://www.linuxfoundation.org/resources/open-source-guides/creating-an-open-source-program. [Accessed: Sep. 05, 2023].

“Open Source Program Offices: A New Era for Open Source,” GitHub, 2019. [Online]. Available: https://www.openoffice.org/ . [Accessed: Sep. 05, 2023].

C. DiBona and J. L. Gaudiot, “Open source software and Google,” IEEE Computer, vol. 43, no. 10, pp. 120-123, Oct. 2010.

S. Kulkarni and S. Ramaswamy, “Microsoft’s journey to becoming an open source company,” IEEE Software, vol. 37, no. 2, pp. 16-21, Mar./Apr. 2020.

J. Pearce, “Facebook’s open source program office,” IEEE Software, vol. 35, no. 4, pp. 14-17, Jul./Aug. 2018.

A. Economize, “What is Augmented FinOps? 5 Greatest Benefits & Examples,” Economize, Sep. 1, 2023. [Online]. Available: https://blog.economize.cloud/augmented-finops/. [Accessed: Sep. 05, 2023].

Tenacity, “Augmented FinOps: Revolutionizing Cloud Financial Management for the Future,” Tenacity, Aug. 23, 2023. [Online]. Available: https://www.tenacity.ai/blog/augmented-finops-revolutionizing-cloud-financial-management-for-the-future. [Accessed: Sep. 05, 2023].

S. Arai, “Column: “Augmented FinOps”,” Alphaus Blog, Aug. 30, 2022. [Online]. Available: https://alphaus.cloud/en/blog/column-augmented-finops-in-gartner-2022-hype-cycle/. [Accessed: Sep. 05, 2023].

“Cloud Development Environments: The Future of Software Development - Codeanywhere,” Codeanywhere, Oct. 29, 2022. [Online]. Available: https://blog.codeanywhere.com/cloud-development-environments-the-future-of-software-development/. [Accessed: Sep. 05, 2023].

“Cloud Workstations | Google Cloud,” Google Cloud, 2021. [Online]. Available: https://cloud.google.com/workstations. [Accessed: Sep. 05, 2023].

“What is Cloud Development Environment,” CloudShare, 2021. [Online]. Available: https://www.cloudshare.com/virtual-it-labs-glossary/cloud-development-environment/. [Accessed: Sep. 05, 2023].

“Cloud Sustainability - The FinOps Alliance”, The FinOps Alliance, 2021. [Online]. Available: https://finopsalliance.com/blog/cloud-sustainability/. [Accessed: Sep. 05, 2023].

Sustainability | Google Cloud”, Google Cloud, 2021. [Online]. Available: https://cloud.google.com/sustainability/. [Accessed: Sep. 05, 2023].

“Microsoft Cloud for Sustainability | Microsoft”, Microsoft, 2021. [Online]. Available: https://www.microsoft.com/en-us/sustainability/cloud. [Accessed: Sep. 05, 2023].

“Green data centers: towards a sustainable digital transformation - A practitioner’s guide”, ITU, 2021. [Online]. Available: https://www.itu.int/en/ITU-D/Environment/Pages/Toolbox/Green-data-center-guide.aspx. [Accessed: Sep. 05, 2023].

"Corporate Social Responsibility (CSR) Explained With Examples", Investopedia, 2021. [Online]. Available: https://www.investopedia.com/terms/c/corp-social-responsibility.asp. [Accessed: Sep. 05, 2023].

"What Is Corporate Social Responsibility? 4 Types | HBS Online", Harvard Business School Online, 2021. [Online]. Available: https://online.hbs.edu/blog/post/types-of-corporate-social-responsibility. [Accessed: Sep. 05, 2023].

"What Is Cloud Native | Google Cloud", Google Cloud, 2021. [Online]. Available: https://cloud.google.com/learn/what-is-cloud-native. [Accessed: Sep. 05, 2023].

"What is Cloud Native? - Everything you need to know - AWS", AWS, 2021. [Online]. Available: https://aws.amazon.com/what-is/cloud-native/. [Accessed: Sep. 05, 2023].

"Cloud-Native Applications 101: Examples, Benefits & Methods - Mendix", Mendix, Oct. 6, 2022. [Online]. Available: https://www.mendix.com/blog/why-cloud-native-applications-are-the-future-of-software/. [Accessed: Sep. 05, 2023].

A. Reznik et al., "Cloud-out and edge-in: How cloud service providers can leverage the two edge computing architectures," Gartner, Stamford, CT, USA, Rep. G00740404, Jul. 2021. [Online]. Available: https://www.gartner.com/en/documents/4003564. [Accessed: Sep. 05, 2023]

"WebAssembly," Mozilla Developer Network, 2021. [Online]. Available: https://developer.mozilla.org/en-US/docs/WebAssembly. [Accessed: 05-Sep-2023].

"WebAssembly," WebAssembly.org, n.d. [Online]. Available: https://webassembly.org/. [Accessed: 05-Sep-2023].

"WebAssembly - Overview," Tutorialspoint.com, 2021. [Online]. Available: https://www.tutorialspoint.com/webassembly/webassembly_overview.htm. [Accessed: 05-Sep-2023].

L. Perri, "Safe and Effective Implementation of AI TRiSM," Gartner, Oct. 19, 2022. [Online]. Available: https://www.gartner.com/en/articles/what-it-takes-to-make-ai-safe-and-effective. [Accessed: Sep. 05, 2023].

L. Siddiqui, "AI TRiSM Explained: AI Trust, Risk & Security Management," Splunk, Mar. 28, 2023. [Online]. Available: https://www.splunk.com/en_us/blog/learn/ai-trism-ai-trust-risk-security-management.html. [Accessed: Sep. 05, 2023].

R. Kaggarwal, "AI TRiSM — Balancing AI Trust, Risk, and Security," Medium, Mar. 04, 2023. [Online]. Available: https://medium.com/@rajneeshkaggarwal/ai-trism-balancing-ai-trust-risk-and-security-f970ff06703a. [Accessed: Sep. 05, 2023].

"What is Cybersecurity Mesh Architecture, CSMA Overview," Grip Security, 2021. [Online]. Available: https://www.grip.security/glossary/cybersecurity-mesh-architecture. [Accessed: Sep. 05, 2023].

"Cybersecurity Mesh Architecture (CSMA) - Check Point Software," Check Point Software, Oct. 18, 2021. [Online]. Available: https://www.checkpoint.com/cyber-hub/cyber-security/what-is-cybersecurity-mesh-architecture-csma/. [Accessed: Sep. 05, 2023].

"Carrier Sense Multiple Access (CSMA) - Scaler Topics," Scaler Topics, Nov. 14, 2022. [Online]. Available: https://www.scaler.com/topics/csma/. [Accessed: Sep. 05, 2023].

"Generative AI: What Is It, Tools, Models, Applications and Use Cases," Gartner, Jan. 19, 2023. [Online]. Available: https://www.gartner.com/en/topics/generative-ai. [Accessed: Sep. 05, 2023].

“What is ChatGPT, DALL-E, and generative AI? | McKinsey,” McKinsey & Company, Jan. 19, 2023. [Online]. Available: https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-generative-ai. [Accessed: Sep. 05, 2023].

“Infographic: Generative AI Explained by AI - Visual Capitalist,” Visual Capitalist, Feb. 01, 2023. [Online]. Available: https://www.visualcapitalist.com/generative-ai-explained-by-ai/. [Accessed: Sep. 05, 2023].

“What is Generative AI and How Does it Impact Businesses? | BCG,” Boston Consulting Group, n.d. [Online]. Available: https://www.bcg.com/capabilities/artificial-intelligence/generative-ai. [Accessed: Sep. 05, 2023].

“Generative AI: A brief overview of its history and impact,” LXT.AI Blog, Apr. 03, 2023. [Online]. Available: https://www.lxt.ai/blog/generative-ai-a-brief-overview-of-its-history-and-impact/. [Accessed: Sep. 05, 2023].

“Homomorphic Encryption: What Is It, and Why Does It Matter?” Internet Society, Mar. 9, 2023. [Online]. Available: https://www.internetsociety.org/resources/doc/2023/homomorphic-encryption/. [Accessed: Sep. 5, 2023].

“Homomorphic Encryption: Overview, applications, & uses,” Security Focal, Jun. 7, 2021. [Online]. Available: https://www.securityfocal.com/homomorphic-encryption-overview-applications-uses/. [Accessed: Sep. 5, 2023].

National Security Agency, "Quantum Computing and Post-Quantum Cryptography," Aug. 2023. [Online]. Available: https://media.defense.gov/2021/Aug/04/2002821837/-1/-1/1/Quantum_FAQs_20210804.PDF. [Accessed: Sep. 05, 2023].

NIST, "NIST Announces First Four Quantum-Resistant Cryptographic Algorithms," Jul. 05, 2022. [Online]. Available: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms. [Accessed: Sep. 05, 2023].

M. Martin, “What is Software Engineering? Definition, Basics, Characteristics,” Guru99, 12-Aug-2023. [Online]. Available: https://www.guru99.com/what-is-software-engineering.html. [Accessed: 05-Sep-2023].

“Software Engineering: What It is, Definition, Tutorial,” javatpoint. [Online]. Available: https://www.javatpoint.com/software-engineering. [Accessed: 05-Sep-2023].